Express Mail Label No. EV396081749US

[0001] # METHOD AND APPARATUS FOR GENERATING COMPLEX FOUR-PHASE SEQUENCES FOR A CDMA COMMUNICATION SYSTEM

[0002] ## CROSS REFERENCE TO RELATED APPLICATIONS

[0003] This application is a continuation of Application No. 10/066,968, filed February 4, 2002; which is a continuation of U.S. Patent No. 6,606,344, which issued on August 12, 2003; which is a continuation of U.S. Patent No. 6,337,875, which issued on January 8, 2002; which is a continuation of U.S. Patent No. 6,026,117, which issued on February 15, 2000.

[0004] ## FIELD OF THE INVENTION

[0005] The present invention generally relates to an improved sequence design for code-division multiple access (CDMA) communications. More particularly, the invention is directed to generating complex four-phase pseudo-random code sequences which may be directly mapped to a quadrature phase shift keying (QPSK) signal constellation.

[0006] ## BACKGROUND

[0007] Code-division multiple access (CDMA) is a type of spread spectrum communication system wherein each subscriber unit is distinguished from all other subscriber units by the possession of a unique code. In order to communicate with a particular subscriber unit, a transmitting unit imprints the unique code upon a transmission and the receiving unit uses the code to decode the transmission. CDMA communication systems transmit voice and data information using signals that appear noiselike and random. Since the random sequences are generated by standard deterministic logic elements, the generation of the bit sequences are predictable and repeatable. It is the use of these repeatable binary random sequences that permits easy modulation of any information-bearing digital signal

for data communications. These predictable random sequences are called pseudo-random sequences.

[0008] Each subscriber unit in a CDMA communication system receives a plurality of pseudo-random sequences from base stations which are within the communicating range of the subscriber unit. As indicated above, the receiving unit uses a particular pseudo-random code to attempt to decode one of the received pseudo-random sequences. The particular code can only be used to decode one pseudo-random sequence, the other received pseudo-random sequences contribute to noise.

[0009] As the correlation between the pseudo-random sequences used by the CDMA communication system decreases, the amount of noise output by the receiving unit also decreases. This decrease can be explained as follows: There is a high correlation between the one pseudo-random sequence including the data to be transmitted to the subscriber unit and the pseudo-random sequence generated by the receiver. As the correlation between the one pseudo-random sequence and the other pseudo-random sequences decreases (i.e. cross correlation), it becomes easier for the subscriber unit to recognize its particular pseudo-random sequence and filter out all of the other pseudo-random sequences. Thus, noise is reduced and signal clarity enhanced.

[0010] There is a need for an improved pseudo-random sequence generator which generates sequences having improved cross correlation properties to reduce the noise experienced by the receiver. There is also a need for a pseudo-random code generator that is easy to implement.

[0011] SUMMARY

[0012] A transmission apparatus for generating a complex four-phase pseudo-random sequence having I and Q portions includes a shift register and an accumulator. The shift register has a plurality of positions. The accumulator has a first input for receiving an output from the shift register and a second input for

receiving a predetermined value. The accumulator combines the data received via the first and second inputs and outputs the combined data to the shift register. Bits from a first predetermined position within the shift register are used to generate the I portion of the sequence and bits from a second predetermined position within the shift register are used to generate the Q portion of the sequence.

[0013] In one embodiment, a pseudo-random code generator produces complex four-phase CDMA codes utilizing an accumulator and a plurality of flip flops. The accumulator receives a quotient of a parameter M divided by a parameter N and receives feedback from the plurality of flip flops. The parameter M and N are integers, wherein M is relatively prime to N. The accumulator combines the quotient with the data received from the flip flops and transmits the combined data to the flip flops. Two bits are extracted and used to produce I and Q codes.

[0014] In another embodiment, a pseudo-random code generator produces complex four-phase CDMA codes by providing a circuit for outputting an arithmetic progression of values and an incremental value of the arithmetic progression of values. The pseudo-random code generator also contains a first mixer for receiving the arithmetic progression of values and the incremental values. A second mixer receives the output of the first mixer and combines this output with the quotient of a parameter 2M divided by parameter N, wherein M and N are integers and M is relatively prime to N. Two bits are extracted from the second mixer and are converted into I and Q codes.

[0015] Other advantages will become apparent to those skilled in the art after reading the detailed description of the preferred embodiments.

[0016] BRIEF DESCRIPTION OF THE DRAWING(S)

[0017] Figure 1 is a block diagram of a spread spectrum transmitter of the present invention.

[0018] Figure 2 is a block diagram of a spread spectrum receiver of the present invention.

[0019] Figure 3 is a timing diagram of a conventional pseudo-random code sequence.

[0020] Figure 4 is a first embodiment of a spread spectrum code generator for generating four-phase sequences according to the present invention.

[0021] Figure 5 is a diagram showing the conversion to I and Q in the first embodiment of the spread spectrum code generator.

[0022] Figure 6 is a diagram showing the method steps for generating four-phase sequences according to the first embodiment of the present invention.

[0023] Figure 7 is a second embodiment of a spread spectrum code generator for generating four-phase sequences according to the present invention.

[0024] Figure 8 is a diagram showing the conversion to I and Q in the second embodiment of the spread spectrum code generator.

[0025] Figure 9 is a diagram showing the method steps for generating four-phase sequences according to the second embodiment of the present invention.

[0026] Figure 10 is a graph of an example of an autocorrelation function for the first suboptimum implementation.

[0027] Figure 11 is an example of a cross correlation function for the first suboptimum implementation.

[0028] DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT(S)

[0029] The preferred embodiments are described with reference to drawing figures wherein like numerals represent like elements throughout.

[0030] A spread spectrum transmitter 10, as shown in Figure 1, includes an analog-to-digital (A/D) converter 12 for receiving a voice signal. A switch 14 receives both the digital voice signal from the A/D converter 12 and a digital data signal from a terminal (not shown). The switch 14 connects the spread spectrum transmitter 10 with an input for either digital voice signal or digital data. The digital voice signal and digital data are hereafter collectively referred to as digital data. The switch 14 directs the digital data to a spreader 20, which may comprise a

mixer. A pseudo-random sequence generated by code generator 30 is applied to the spreader 20. The code generator 30 and the spreader 20 are shown as being contained within spread spectrum encoder 40.

[0031] The spreader 20 performs a frequency spectrum spreading function by multiplying the digital data by the pseudo-random sequence in the time domain, which is equivalent to convolving the bimodal spectrum of the digital data with the approximately rectangular spectrum of the pseudo-random sequence in the frequency domain. The output of the spreader 20 is applied to a low-pass filter 50, whose cutoff frequency is equal to the system chip rate, F_{cr}. The output of the low-pass filter 50 is then applied to one terminal of a mixer 60 and upconverted, as determined by the carrier frequency F_c which is applied to its other terminal. The upconverted signal is then passed through a band-pass filter 70, which may be a helical resonator. The filter 70 has a bandwidth equal to twice the chip rate and a center frequency equal to the center frequency of the bandwidth of the spread spectrum system. The output of the filter 70 is applied to the input of an RF amplifier 80, whose output drives an antenna 90.

[0032] A spread spectrum receiver 100 is shown in Figure 2. An antenna 110 receives the transmitted spread spectrum signal, which is filtered by a bandpass filter 120. The filter has a bandwidth equal to twice the chip rate F_{cr}, and a center frequency equal to the center frequency of the bandwidth of the spread spectrum system. The output of the filter 120 is subsequently downconverted by a mixer 130, possibly in two stages, to a baseband signal using a local oscillator having a constant frequency which is approximately the same as the carrier frequency F_c of the transmitter 10. The output of the mixer 130 is then despread by applying it to a first terminal of the despreader 140 while applying the same pseudo-random sequence as delivered to the spreader 20 to a second terminal of the despreader 140. The pseudo-random sequence is generated by a code generator 30. The despreader 140 and the code generator 30 are contained within a spread spectrum decoder 160 as shown in Figure 2. The output of the despreader 140 is applied to a low pass

filter 180, which has a cutoff frequency at the data rate of the data input to the spread spectrum transmitter 10. The output of the low-pass filter 180 is a replica of the data input to Figure 1.

[0033] It should be appreciated by those of skill in the art that the pseudo-random sequence used in the receiver 100 of a spread spectrum communication system must be synchronized with the pseudo-random sequence used in the transmitter 10. Methods for achieving this synchronization are also well known.

[0034] A conventional spreading sequence is a pseudo-random digital sequence as shown in Figure 3. The sequence is used to spread the signal being transmitted and to despread the signal being received. Two different binary codes using two different LFSR circuits provide I and Q channels for transmission of data. However, if there is high cross-correlation between the I and Q channels at the receiver side, a great deal of noise will be output by the receiver.

[0035] The code generator 30 of the present invention generates pseudo-random code sequences with greatly enhanced cross-correlation properties compared with the prior art pseudo-random sequences such as the one shown in Figure 3. A prior art pseudo-random sequence essentially comprises a signal having different frequency components. This signal is a combination of sinusoidal waveforms having different frequencies; both high frequency sinusoidal waveforms and low frequency sinusoidal waveforms. Thus, the signal has a frequency spectrum which can be divided into frequency regions. Those sinusoids having stronger frequencies (higher amplitudes) will be more dominant in the signal than those sinusoids having weaker frequencies (lower amplitudes). However, in order to generate an enhanced pseudo-random code (highly random code) as in the present invention, the strength or amplitude in each frequency region should be the same. Highly random codes have the property that they contain components in all frequency regions, resulting in a flat spectrum. The code generator 30 generates a pseudo-random sequence wherein the amplitude of the sinusoids in all frequency regions is approximately the same (flat) as will be explained in detail below.

[0036] A pseudo-random sequence having a length N and frequency regions X can be represented by Y frequency bins of a discrete Fourier series representation, wherein each bin corresponds to a frequency region. There are Y bins for the X frequency regions $(2\pi/T)k$, k = 0,..., N-1 where T is the period of the spreading sequence in time and X=Y=N. The instantaneous frequency of the sequence should ideally spend equal time in each of the X frequency regions. Therefore, each frequency region or bin will have the same strength. For example, let s(t) denote the spreading sequence which is periodic. Then

$$s(t) = \sum_k c_k e^{j2\pi kt/T} \qquad \text{Equation (1)}$$

is the Fourier Series representation where

$$c_k = \frac{1}{T}\int_T s(t) e^{-j2\pi kt/T} dt \qquad \text{Equation (2)}$$

where c_k is the strength of the sinusoids at one of the discrete Fourier series representations or the strength of the sinusoids in the region or bin. The average power in s(t) is written as follows:

$$P=\sum_K |c_k|^2 \qquad \text{Equation (3)}$$

The magnitude spectrum of s(t) is $|c_k|$ and power spectrum is $|c_k|^2$. The ideal power spectrum is flat, where the average power is distributed over all frequency bins equally. This results in a narrow autocorrelation. All of the $|c_k|^2$ should be equal. To obtain this, the instantaneous frequency is:

$$\frac{2\pi}{T} Mk, \quad k = 0,..., \quad N - 1 \qquad \text{Equation (4)}$$

where M and N are integers and M is relatively prime to N (M and N do not have the same common factor). This guarantees that each frequency bin $(2\pi/T)k$ is visited equally. For example, if N=7 and M=3, the instantaneous frequency is then

$$0, \frac{2\pi}{T}x3, \frac{2\pi}{T}x6, \ldots, \frac{2\pi}{T}x18 \qquad \text{Equation (5)}$$

Since a discontinuity in the phase has the effect of spreading the power into other frequency bins, the phase is preferably continuous and free of sudden bumps as much as possible.

[0037] The primary constraint is that the phase of the complex spreading sequence should be limited to {0, π/2, π, 3π/2}. This limitation leads to sudden phase changes and prevents the power spectrum from becoming completely flat. However, a sequence with relatively flat power spectral density can be obtained. For the phase to be continuous at t = (k/N)T, the recursive equation is

$$\Theta_{k-1} - \Theta_k = \frac{2\pi}{N}Mk \qquad \text{Equation (6)}$$

where Θ is the phase of individual chips in a sequence and k is the index (order) of the chips in the sequence. If Θ_0 is arbitrarily chosen as one of (0, π/2, π, 3π/2), then Θ_1, Θ_2, ..., Θ_N can be generated sequentially. This solution results in flat spectra, which is the optimum solution. The choice of Θ_0 (0, π/2, π, 3π/2) makes no difference because a constant phase offset over the sequence does not change its spectral properties.

[0039] The suboptimum implementation of the above equation when Θ_k is limited to {0, π/2, π, 3π/2} is as follows:

$$\Theta_{k-1} - \Theta_k = \frac{\pi}{2}\left(\left\lfloor 4\frac{M}{N}k \right\rfloor \bmod 4\right) \qquad \text{Equation (7)}$$

where $\left\lfloor 4\frac{M}{N}k \right\rfloor$ means the largest integer less than or equal to 4(M/N)k. This equation is a modified version of Equation (6) and it performs the mapping of phase angles to one of four points for easy QPSK implementation. It limits the phases to the set {0, π/2, π, 3π/2}.

[0040] Continuing the sequential phase deviation to develop a second suboptimum implementation, one has:

$$\Theta_k = \Theta_{k-1} - \frac{2\pi}{T} M \frac{k}{N} T$$

Equation(8)

$$\Theta_k = \Theta_{k-2} - \frac{2\pi}{T} M \frac{k-1}{N} T - \frac{2\pi}{T} M \frac{k}{N} T$$

.

.

.

$$\Theta_k = \Theta_0 - \frac{2\pi}{T} M \frac{T}{N} \sum_{i=1}^{k} i = \Theta_0 - \frac{2\pi}{T} M \frac{T}{N} \frac{k(k+1)}{2}$$

Equation (9)

$$\Theta_k = \Theta_0 - \pi \frac{M}{N} k(k+1)$$

Again, the second suboptimum implementation with four phases (0, π/2, π, 3π/2) is obtained as:

$$\Theta_k = \Theta_0 - \frac{\pi}{2} \left(\left\lfloor 2 \frac{M}{N} k(k+1) \right\rfloor \bmod 4 \right)$$

Equation (10)

If $\Theta_0=0$, then:

$$\Theta_k = \frac{\pi}{2} \left\lfloor 2 \frac{M}{N} k(k+1) \right\rfloor \bmod 4$$

Equation (11)

for this second suboptimum implementation.

[0041] Examining Equation 6 one sees that each phase term can be obtained by adding a variable term (2 π/N)(Mk) to the previous phase. Furthermore, since 2πk is equal to zero modulo 2π, the term one needs to add each phase to find the next phase reduces to (M/N), which is not an integer. Therefore, a possible implementation can be a recursive adder (accumulator) which adds the term (M/N) to the phase in each iteration.

[0042] Figure 4 shows a first embodiment of the code generator 30 for generating four-phase pseudo-random code sequences which greatly improve autocorrelation properties and cross correlation properties. The first embodiment is an example of the first suboptimum implementation of Equation 7. Although four-phase sequences of any length can be generated, a length of 127 bits is selected as an example. Further, for the purposes of this example, there are N number of chips in a symbol, which represents the processing gain. A number M is selected to be relatively prime to N, which means that M and N do not have a common factor. The number of bits L required to provide a binary representation of the processing gain N is determined by solving the following equation:

$$N \leq 2^L. \qquad \text{Equation (12)}$$

[0043] The code generator 30 includes an accumulator 31 which is 2L bits in length. Since N=127 in this example, L=8. Therefore, accumulator 31 has a length of 16 bits. An eight bit number M/N is applied to one input of the accumulator 31. A sixteen bit number from flip flops 32_1 through 32_{2L} is applied to a second input for the accumulator 31. Flip flops 32_1 through 32_{2L} may be replaced by a shift register. Although bits are input to flip flops 32_1-32_{2L} and to accumulator 31 in parallel, the bits could also be input in series. The sum of the two numbers input into the accumulator 31 is transmitted to flip flops 32_1 through 32_{2L}. An extractor 33 extracts the fifth and sixth least significant bits from the flip flops 32_1 through 32_{2L} (Figure 5). The fifth and sixth least significant bits are applied to an exclusive-or gate 34.

[0044] The output of the exclusive-or gate 34 is converted to a Q value by a converter 36. The sixth bit output from extractor 33 is converted to an I value by converter 35. The I and Q values output from converters 35 and 36 are applied to spreader 20 or despreader 140. As indicated before, M/N is an eight bit number in this example. The fifth and sixth bits of the accumulator output represent the first two significant bits of 4 (M/N) which appears in Equation (7). When 4 (M/N) is mapped to one of four values {0, 1, 2, 3} by taking modulo 4, the result is the first two significant bits of 4(M/N), or equivalently fifth and sixth bits of the accumulator.

[0045] Figure 6 is a flow diagram of the method performed by the circuit shown in Figure 4. The initial parameters M and N are loaded into registers or memory (not shown) before performing the dividing function (M divided by N). In addition, the value in accumulator 31 is preferably equal to zero. The remaining apparatus in the code generator 30 is also initialized (S1). The sum, which initially is zero, is added to the quotient of M/N (S2). The fifth and sixth bits of the new sum are extracted (S3) in order to be converted into the I and Q values (S4 and S5). The bits (L-2) and (L-3) should be mapped to QPSK constellation as follows:

$$00 \rightarrow 11$$

$$01 \rightarrow 1\text{-}1$$

$$10 \rightarrow -1\text{-}1$$

$$11 \rightarrow -11$$

This mapping can be done in software or hardware by using first:

(L-2)	(L-3)		(L-2)	(L-2) ⊕ (L-3)
0	0	→	0	0
0	1	→	0	1
1	0	→	1	1
1	1	→	1	0

and then using the standard 0→1, 1→-1 mapping.

[0046] For example, if the sixth bit for L-2 bit is equal to zero, then the I value is one. If the sixth bit is a one, then the I value is negative one. In the case of the Q value, if the output of exclusive-or gate 34 is a zero, the Q value is one. If the output of exclusive-or gate 34 is a one, the Q value is negative one. The I and Q values are output to the spreader 20 or despreader 140 (S6). Method steps S2 through S6 are repeated until all the digital data supplied by switch 14 is transmitted or all the data is received by switch 190.

[0047] Figure 7 shows a second embodiment of the code generator 200. Code generator 200 is substituted for code generator 30 and generates four-phase pseudo-random code sequences similar to those generated by the code generator 200 which greatly improve auto correlation properties and cross correlation properties. The second embodiment is an example of the second suboptimum implementation of Equation (11). Although four-phase sequences of any length can be generated, a length of 127 bits is selected as an example. Further, for the purposes of this example, there are N number of chips in a symbol, which represents the processing gain. A number M is selected to be relatively prime to N. The number of bits L required to provide a binary representation of processing gain N is determined by solving Equation (12). Since M=127 in this example, L=8. Therefore (M/N) is sixteen bits in length.

[0048] The code generator 30 includes an accumulator 210 which is L bits in length. Accumulator 210 has a length of 8 bits. A "1" is preferably applied to one input of accumulator 210. The number from flip flops 220_1 through 220_L is applied to a second input of the accumulator 210. Flip flops 220_1 through 220_L may be replaced by a shift register. Although bits are input to flip flops 220_1 through 220_L and accumulator 210 in parallel, the bits could be input in series. The sum of the two numbers input into the accumulator 210 is transmitted to flip flops 220_1 through 220_L. The output of flip flops 220_1 through 220_L are transmitted to flip flops 230_1 through 230_L as well as mixer 240. The mixer 240 also receives the

output of flip flops 230_1 through 230_L. The accumulator 210 and flip flops 220_1-220_L, flip flops 230_1-230_L, and mixer 240 provide a flip flop feedback circuit. The output of mixer 240 is input to mixer 250. Mixer 250 also receives an 8 bit input from (M/N). The extractor 260 extracts the fifth and sixth least significant bits from the mixer 250. The sixth least significant bit output from extractor 260 is converted to an I value by converter 280. The fifth and sixth least significant bits are applied to an exclusive-or gate 270. The output of the exclusive-or gate 270 is converted to a Q value by a converter 290 as shown in Figure 8. The I and Q values output from converters 280 and 290 are applied to spreader 20 or despreader 140. As indicated before, (M/N) is an eight bit number in this example. Flip flops 220_1 through 220_l output the k value and flip flops 230_1 through 230_L output the k+1 value to the mixer 240. The mixer 250 receives the output of mixer 240 and the product of (M/N). When 2(M/N)k(k+1) is mapped to one of the four values {0, 1, 2, 3} by taking modulo 4, the result is the fifth and sixth bits from extractor 260 (Figure 8).

[0049] Figure 9 is a flow diagram of the method performed by the circuit shown in Figure 7. The initial parameters M and N are loaded into registers or memory (not shown) before performing the dividing function (M/N). In addition, the value k is preferably equal to zero. The remaining apparatus in the second embodiment of the code generator 200 is also initialized (S1). The value of (M/N)k(k+1) is calculated (S2). The fifth and sixth bits resulting from the above calculation are extracted (S3) in order to be converted into I and Q values (S4 and S5). The bits (L-2) and (L-3) should be mapped to QPSK constellation as follows:

$00 \rightarrow 11$

$01 \rightarrow 1\text{-}1$

$10 \rightarrow \text{-}1\text{-}1$

$11 \rightarrow \text{-}11$

This mapping can be done in software or hardware by using first:

(L-2)	(L-3)		(L-2)	(L-2) ⊕ (L-3)
0	0	→	0	0
0	1	→	0	1
1	0	→	1	1
1	1	→	1	0

and then using the standard 0→1, 1→-1 mapping.

[0050] For example, if the sixth bit for L-2 is equal to zero, then the I value is 1. If the sixth bit is a 1, then the I value is -1. In the case of the Q value, if the output of the exclusive-or gate 270 is a zero, the Q value is 1. If the output of the exclusive-or gate 270 is a 1, the Q value is -1. The I and Q values are output to the spreader 20 or the despreader 140 (S6). The k value is incremented. Method steps S2 through S7 are repeated into all the digital data supplied by switch 14 is transmitted where all the data is received by switch 190.

[0051] Figure 10 shows an auto correlation function where N=127 and M=44, which is the result of using the first suboptimum implementation to generate the pseudo-random code.

[0052] Figure 11 shows a cross correlation function where N=127 and M=44, which is the result of using the first suboptimum implementation to generate the pseudo-random code.

[0053] The autocorrelation a(n) for the sequence s(k) is given as:

$$a(n) = \sum_{k=1}^{N} s(k)s^{*}(k+n) \qquad \text{Equation (13)}$$

where the indexes in parentheses are taken modulo N, and the cross correlation c(n) of two sequences s(k) and r(k) is given as:

$$c(n) = \sum_{k=1}^{N} s(k) r^{*}(k+n) \qquad \text{Equation (14)}$$

where again the index is taken modulo N. The first suboptimum implementation achieves the desirable result of making the magnitude of the cross correlation and autocorrelation (except for a(0)) small compared to N. Although the results of the example of the second suboptimum implementation are not shown, the results are similar. Equations 13 and 14 are well known to one having ordinary skill in the art.

[0054] Although the invention has been described in part by making detailed reference to certain specific embodiments, such detail is intended to be instructive rather than restrictive. It will be appreciated by those skilled in the art that many variations may be made in a structure and mode of operation without departing from the spirit and scope of the invention as disclosed in the teachings herein.

* * *